#2
8-21-01

日 本 国 特 許 庁
PATENT OFFICE
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２０００年　５月１８日

出願番号
Application Number: 特願２０００－１４６８６７

出願人
Applicant (s): 株式会社日立製作所

U. S. Appln. Filed 1-9-01
Inventor: M. Asano et al
Mathingly Stanger & Malur
Docket ASA-954

２０００年１２月１５日

特許庁長官
Commissioner,
Patent Office 及川耕造

出証番号　出証特２０００－３１０４４０９

【書類名】　特許願

【整理番号】　KN1113

【提出日】　平成12年 5月18日

【あて先】　特許庁長官殿

【国際特許分類】　G06F　9/46

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　浅野　正靖

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　新井　利明

【発明者】

【住所又は居所】　神奈川県横浜市戸塚区戸塚町５０３０番地　株式会社日立製作所　ソフトウェア事業部内

【氏名】　山下　洋史

【特許出願人】

【識別番号】　000005108

【氏名又は名称】　株式会社　日立製作所

【代理人】

【識別番号】　100078134

【弁理士】

【氏名又は名称】　武　顕次郎

【電話番号】　03-3591-8550

【手数料の表示】

【予納台帳番号】　006770

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　計算機システム及び計算機システムの制御方法

【特許請求の範囲】

【請求項１】　複数のオペレーティングシステムを１台の計算機内で稼動させ、ＣＰＵ、主記憶装置、外部入出力装置等の計算機資源を前記複数のオペレーティングシステムのそれぞれに割り当てる手段を有する計算機システムにおいて、前記計算機資源を管理する手段と、前記計算機資源の前記各オペレーティングシステムへの割り当ての変更と再構成とを行う手段と、前記各オペレーティングシステムの稼動状態に関連して前記計算機資源の変更と再構成との内容を管理する手段と、前記各オペレーティングシステムの稼動状態に基づいて前記計算機資源を変更または再構成する手段とを備えることを特徴とする計算機システム。

【請求項２】　複数のオペレーティングシステムを稼動させ、オペレーティングシステムの１つを運用系オペレーティングシステムとし、他の１つを待機系オペレーティングシステムとして、運用系オペレーティングシステムに障害が生じたとき、待機系オペレーティングシステムに処理を継承させるクラスタシステムを１台の計算機内に構成した計算機システムの制御方法において、運用系オペレーティングシステムの障害を監視し、かつ、待機系オペレーティングシステムに処理を継承する動作を監視して、運用系オペレーティングシステムの稼動時、運用系オペレーティングシステムに待機系よりも多くの計算機資源を割り当てておき、運用系オペレーティングシステムに障害が発生して、待機系オペレーティングシステムが処理を開始した場合、待機系オペレーティングシステムに計算機資源を多く割り当てて、待機系オペレーティングシステムを運用系として動作させることを特徴とする計算機システムの制御方法。

【請求項３】　複数のオペレーティングシステムを１台の計算機内で稼動させている計算機システムの制御方法において、前記オペレーティングシステムの稼動状態を監視して各ＯＳの負荷を監視すると共に、この負荷を分析し、高い負荷を持つオペレーティングシステムに関する負荷の原因を特定し、前記原因を解決するために必要な計算機資源を多く割り当てることを特徴とする計算機システムの制御方法。

【請求項４】　複数のオペレーティングシステムを１台の計算機内で稼動させている計算機システムの制御方法において、複数のオペレーティングシステムのそれぞれが扱う処理群を管理し、前期処理を他のオペレーティングシステムの処理と比較して優先的に処理を行う必要がある場合、優先的に処理を行う必要があるオペレーティングシステムに計算機資源を多く割り当てることを特徴とする計算機システムの制御方法。

【請求項５】　複数のオペレーティングシステムを１台の計算機内で稼動させている計算機システムの制御方法において、複数のオペレーティングシステムのそれぞれが扱う処理群と各処理を実行する稼動時間とを管理し、稼働時間に従って計算機資源の変更内容と有効期間とを管理し、オペレーティングシステムの扱う処理と計算機資源の動的割り当てとを関連させて、各オペレーティングシステムに対する計算機資源の割り当てを行うことを特徴とする計算機システムの制御方法。

【請求項６】　複数のオペレーティングシステムのいずれにも割り当てられていない計算機資源を管理し、各オペレーティングシステムに割り当てられている計算機資源の使用率を管理し、この使用率に従って各オペレーティングシステムに、割り当てられていない計算機資源を割り当てることを特徴とする請求項３、４または５記載の計算機システムの制御方法。

【請求項７】　計算機システムに接続して計算機資源を使用する使用者を管理し、使用者の計算機資源の使用時間を管理し、使用者の要求により計算機資源の割り当てを変更し、使用者が設定した計算機資源の割り当てを行い、その割り当ての状態による課金を行うことを特徴とする請求項６記載の計算機システムの制御方法。

【請求項８】　複数のオペレーティングシステムを１台の計算機内で稼動させている計算機システムの制御方法において、複数のオペレーティングシステムに共通の主記憶領域を確保し、前記主記憶領域に通信用の形態を形成するデータをオペレーティングシステムが書き込みかつ読み出すことにより、前記オペレーティングシステム相互間で擬似的に通信媒体を形成することを特徴とする計算機システムの制御方法

【請求項９】　複数のオペレーティングシステムを１台の計算機内で稼動させている計算機システムの制御方法において、前記各オペレーティングシステムが独立に稼動しており、共有資源を用いて、各オペレーティングシステムの動作を監視することを特徴とする計算機システムの制御方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、１台の計算機上で複数のオペレーティングシステム（以下、ＯＳという）が動作する計算機システム及び計算機システムの制御方法に係り、特に、複数のＯＳの稼動状態において、計算機資源を効率的に運用することを可能にした計算機システム及び計算機システムの制御方法に関する。

【０００２】

【従来の技術】

計算機の信頼性向上、負荷分散のための制御に関する従来技術として、例えば、特開平１１－３５３２９２号公報等に記載された技術が知られている。この従来技術は、クラスタシステムと呼ばれるものであり、計算機上で動作するあるＯＳが障害を起こした場合、別のＯＳが障害を起こしたＯＳの処理を引き継いで処理を継続することを可能としたシステムである。これにより、この従来技術のシステムは、計算機上のＯＳに障害が発生した場合にも、計算機上で稼動しているプログラムを継続して実行することが可能となる。

【０００３】

【発明が解決しようとする課題】

前述した従来技術による計算機システムであるクラスタシステムは、複数のＯＳを稼動させるために、クラスタを構成するＯＳの数と同数の計算機が必要である。また、従来のクラスタシステムは、運用系の計算機と待機系の計算機とを備えて、かつ、それぞれの計算機が独自に動作している。なお、運用系の計算機とは、通常の動作している計算機、待機系の計算機とは、運用系の計算機が障害となったとき、運用系で操作している処理を引き継ぐために必要な計算機である。

このため、従来技術によるクラスタシステムは、運用系の計算機に障害が起き

た場合、待機系の計算機に処理が引き継がれるが、待機系の計算機で独自に動作しているプロセスがある場合、運用系の計算機の全処理能力を引き継ぐことはできないという問題点を有している。また、従来技術のシステムは、運用系の計算機の処理能力を引き継ぐために待機系の動作を止めておくこととすると、待機系の計算機が使用されず、計算機資源を効率的に使用することができないという問題点を生じる。

【０００４】

例えば、２台の計算機でクラスタシステムを構成した場合、このシステムは、１台の計算機を運用系の計算機、もう１台の計算機を待機系の計算機として構成されるが、運用系の計算機が障害となったとき、その処理能力の全てを待機系の計算機が引き継ぐには、待機系の計算機が、運用系の計算機と同一の処理能力を持ち、かつ、運用系の計算機に障害が起ったとき、待機系の計算機で動作している処理を停止して、運用系の計算機から引き継いだ処理を行わなければならないことになる。また、待機系の計算機で動作している処理停止せずに、運用系の計算機から引き継いだ処理を行うためには、待機系の計算機として、運用系の計算機の処理能力以上の計算機を用いなければならないことになる。

【０００５】

すなわち、従来技術によるクラスタシステムは、通常の運用系の計算機１台の処理能力を達成するために、運用系の計算機以上の処理能力を持つ計算機を２台備えなければならないという問題点を有している。

【０００６】

本発明の目的は、前述した従来技術の問題点を解決し、計算機の信頼性の向上図るためのクラスタシステムを構成しながら、計算機資源の使用の無駄を無くして効率的な運用を行うことのできる計算機システム及び計算機システムの制御方法を提供することにある。

【０００７】

【課題を解決するための手段】

本発明によれば前記目的は、複数のＯＳを１台の計算機システム内で動作させ、各ＯＳの状態を監視して、計算機資源を動的に変更、再配置を行うことにより

、計算機システム上のＯＳの状態により、計算機資源の効率的な運用を可能とすることにより達成される。例えば、１台の計算機システム上で動作する１つのＯＳを運用系ＯＳ、他の１つを待機系ＯＳとしてクラスタシステムを構成し、運用系のＯＳに計算機資源を多く割り当て、運用系ＯＳに障害が起った場合、待機系ＯＳに運用系ＯＳの処理を引き継がせ、そのとき、計算機資源を新たに運用系となる待機系ＯＳに多く割り当てるようにする。これにより、計算機資源を効率よく運用することが可能となる。

【０００８】

また、前記目的は、計算機資源の変更、再配置の環境を用いて、ＯＳの障害のみではなく、様々な状況を監視して、その状況に従い計算機資源を効率よく運用するようにすることにより達成される。例えば、あるＯＳの処理数の増加に伴うＯＳの負荷の増大に対して、そのＯＳに計算機資源を多く割り当てることにより、ＯＳの処理能力を向上させ、負荷の減少につなげることが可能である。また、複数のＯＳの全ての負荷が増大した場合、待機させておいたＣＰＵやメモリを加えて、処理能力を向上させ、終了すれば、その資源を解放することにより、計算機資源の効率的な運用を図ることもできる。

【０００９】

また、前記目的は、計算機資源の再配置に従い、計算機システムを使用する複数のユーザが、特定の状況で多くの資源を使いたい場合、それを計算機システムの管理側に登録し、計算機の資源の使用によって、使用ユーザに使用計算機資源の割合ごとに課金を行うようにすることにより達成される。これにより、計算機システムをサーバとして複数のユーザに展開することが可能となり、また、その使用の柔軟さと課金システムとを提供することができる。

【００１０】

また、前記目的は、１台の計算機システム内で複数のＯＳを稼動させる環境を作り出すＯＳ管理制御部が全てのＯＳを管理できるようにし、複数のＯＳのために共有の主記憶領域を提供して、その共有の主記憶領域を用いて、外部入出力装置によらずに、主記憶経由でＯＳ間通信を行わせるようにすることにより達成される。この通信手段を用いてネットワーク等と同等の通信用プロトコルを作り出

して、共有メモリの送受信を行うことにより、特別なプロトコルを採用することなく、擬似的に外部入出力装置を主記憶上に実現することが可能となる。

【００１１】

【発明の実施の形態】

以下、本発明による計算機システム及び計算機システムの制御方法の実施形態を図面により詳細に説明する。

【００１２】

図１は本発明の第１の実施形態による計算機システムの構成を示すブロック図、図２は計算機システム上で動作する各ＯＳをクラスタ構成とした場合の概念的な構成を示すブロック図、図３は計算機の状態を管理するテーブルの構成を説明する図、図４はＯＳの障害時の計算機資源の変更処理の動作を説明するフローチャートである。図１～図３において、１００は計算機システム、１１０はプロセッサ群、１２０は主記憶装置、１３０は外部入出力装置群、１４０は複数ＯＳ管理制御支援装置、１５０はバス、２００は複数ＯＳ管理制御部、２０１、２０２は第１、第２ＯＳ、２０３、２０４はクラスタサービス、２０５、２０６は監視エージェント、２０７、２０８はアプリケーション、２２１、２２２は第１、第２ＯＳ用ＣＰＵ群、２３０～２３２は端末である。

【００１３】

本発明の第１の実施形態による計算機システム１００は、図１に示すように、プロセッサ群１１０、主記憶装置１２０、外部入出力装置群１３０、複数ＯＳを１台の計算機システム上で動作させるために設けられる複数ＯＳ管理制御支援装置１４０、及び、バス１５０により構成されている。複数ＯＳ管理制御支援装置１４０は、１台の計算機システムが複数のＯＳを実行することができるように、計算機システム１００の構成要素であるプロセッサ群１１０、主記憶装置１２０、外部装置群１３０の仲介をとる装置である。プロセッサ群１１０は、１つ以上のプロセッサの集合であり、プロセッサ（ＣＰＵ）１１ａ、１１ｂ、……、１１ｎからなる。主記憶装置１２０は、複数の各ＯＳ－１～ＯＳ－ｎ用に独立した主記憶領域１２ａ、１２ｂ、……、１２ｎと、複数ＯＳ管理制御部用の主記憶領域１２１と、使用されていない空き領域１２２とにより構成されている。外部装置

群１３０は、入力用のキーボード１３１、出力用ディスプレイ１３２、各ＯＳ用外部記憶装置１３３～１３５、通信装置１３６、１３７、その他の図示しないデバイスから構成されている。

【００１４】

前述のような構成を有する計算機システム１００上で動作する各ＯＳをクラスタ構成とした場合の概念的な構成を示す図２の例は、ＯＳ－１とＯＳ－２と（第１ＯＳ、第２ＯＳともいう）がクラスタ構成を取っている。そして、各ＯＳに割り当てた装置として、外部装置群１３０から、ＯＳ－１用の外部記憶装置１３３、ＯＳ－２用の外部記憶装置１３４、各ＯＳに共有のディスク装置１３５、ＯＳ－１用の通信装置１３６、１３７、ＯＳ－２用の通信装置１３８、１３９が備えられている。前述のＯＳ－１用の通信装置１３６とＯＳ－２用の通信装置１３９との間、及び、ＯＳ－１用の通信装置１３７とＯＳ－２用の通信装置１３８との間は、それぞれケーブル２１１、２１２により接続されている。ケーブル２１１は、他の計算機に接続する外部接続用ケーブルであり、ケーブル２１２は、クラスタを構成するＯＳ－１とＯＳ－２との間のみを接続するクラスタ用接続ケーブルである。ケーブルは、２種類あるが、これはクラスタの信頼性向上のためであり、ケーブル２１２を設けなくてもクラスタ構成を行うことは可能である。さらに、ＯＳ－１、ＯＳ－２のそれぞれには、ＯＳ－１用ＣＰＵ群２２１、ＯＳ－２用ＣＰＵ群２２２が割り当てられている。

【００１５】

前述において、主記憶装置１２０の複数ＯＳ管理制御部領域１２１には、複数ＯＳ管理制御部２００が割り当てられており、複数ＯＳ管理制御部２００は、計算機資源であるプロセッサ群１１０、主記憶装置１２０、外部入出力装置群１３０の構成を変更することが可能であるものとする。また、クラスタに接続される端末がある場合、それらの端末２３０～２３２は、ケーブル２１１に接続される。端末の１つ、例えば端末２３０は、ＯＳ－１、ＯＳ－２の状態を管理し、その状態によって、計算機資源の変更の通知を、ケーブル２１１、各ＯＳを介して行う管理端末としての役割を果たすことも可能である。

【００１６】

主記憶装置１２０内のＯＳ－１領域１２ａ、ＯＳ－２領域１２ｂには、第１ＯＳ２０１、第２ＯＳ２０２がそれぞれ割り当てられ、また、クラスタサービス２０３、２０４、監視エージェント２０５、２０６、各ＯＳで動作するアプリケーション２０７、２０８が格納されている。監視エージェント２０５、２０６は、計算機システムの状態を解析し、計算機資源を効率に運用するために、計算機システムの状態に応じてＯＳ管理制御部２００に計算機資源の変更、再配置を要求するプログラムである。

【００１７】

第１及び第２ＯＳ用ＣＰＵ群２２１、２２２を構成するＣＰＵの割り当て方法としては、次の２種類がある。１つは、ＣＰＵ群１１０の中のＣＰＵ－１とＣＰＵ－２とを第１ＯＳ用に用いるＣＰＵ群２２１とし、ＣＰＵ－３とＣＰＵ－４とを第２ＯＳ用に用いるＣＰＵ群２２２とするというように、各ＣＰＵを個々のＯＳでのみ使用する構成方法である。もう１つは、両方のＣＰＵ群２２０、２２１に同じＣＰＵ１、ＣＰＵ２を割り当てて、各ＣＰＵが時間間隔でそれぞれＣＰＵを各ＯＳに振り分けるて使用する構成方法である。本発明は、これらのどの方法用いて各ＯＳ用のＣＰＵ群を構成してもよい。そして、本発明は、クラスタシステムを構成する複数のＯＳに対して、そのＯＳの状況に応じて、ＣＰＵ、メモリ等の計算機資源を変更可能に割り振ることを可能にしており、例えば、２つのＯＳがクラスタシステムを構成し、その一方が運用系、他方が待機系として動作する場合、運用系のＯＳに対して多くの計算機資源を割り当てるようにしている。この計算機資源の各ＯＳへの割り当ては、計算機資源管理テーブル３００により管理される。

【００１８】

クラスタ構成となっているＯＳ－１とＯＳ－２とに対する計算機資源の割り当てを管理する計算機資源管理テーブル３００は、図３にその例を示すように構成されており、主記憶装置１２０内の図示しない領域に格納されて、複数ＯＳ管理制御支援装置１４０が管理可能であればよい。なお、以後の説明の全てのテーブルについても同様である。計算機資源管理テーブル３００は、計算機資源割り当てテーブル３１０とＯＳ状態テーブル３２０との２つのテーブルにより構成され

ている。このテーブル３００を管理する複数ＯＳ管理制御支援装置１４０内に設けられる図示しない各資源対応の処理実行部は、このテーブルの値を基準に、計算機の状態と計算機資源の状態とを把握して、計算機資源の割り当ての変更を行う。計算機資源割り当てテーブル３１０には、計算資源名３１１と、運用系の計算機に対する計算機資源サービス比３１２と、待機系の計算機に対する計算機資源サービス比３１３とが格納される。これにより、計算機資源とクラスタシステムにおける計算機の役割との関連が設定される。ＯＳ状態テーブル３２０には、ＯＳ名３２１と、各ＯＳの動作状態を示す情報３２２と、各ＯＳが運用系か待機系かを示す情報３２３と、各ＯＳに対する資源（ＣＰＵ）のサービス比率３２４が格納される。このＯＳ状態テーブル３２０により、各ＯＳの動作状況を把握することが可能である。

【００１９】

前述したような構成を持つ計算機資源管理テーブル３００は、クラスタ構成、前述の各資源対応の処理実行部の役割の割り当てによって、テーブルを管理する処理実行部により変更される。図３に示すテーブルにおいては、運用系のＯＳ－１に対して、ＣＰＵ資源の９５％が割り当てられ、残りの５％が、待機系のＯＳ－２に対して割り当てられていることを示している。また、図３には、主記憶装置の割り当て量について記載されていないが、主記憶装置の割り当て量も、運用系のＯＳ－１に対して多くが割り当てられる。なお、主記憶の割り当ては、サービス比率でなくアドレス範囲で行ってもよい。また、計算機資源割り当てテーブル３１０には、ＣＰＵと主記憶とを、資源名として登録しているが、外部記憶装置等についても、その使用台数等について登録しておくことができる。

【００２０】

次に、図４に示すフローを参照して、ＯＳの障害時の計算機資源の変更処理の動作を説明する。この処理は、前述した計算機資源テーブル３００を各監視エージェント２０５、２０６が管理し、クラスタサービス２０３、２０４と、ＯＳ管理制御部１２１と、監視エージェント２０５、２０６とが連携して、運用系のＯＳであるＯＳ－１に障害が起きた場合に計算機資源の変更を行う場合を例とした処理である。

【００２１】

（１）各監視エージェント２０５、２０６は、定期的にクラスタサービス２０３、２０４やＯＳ管理制御部１２１との間で通信を行い、クラスタサービスまたはＯＳ管理制御部から、自監視エージェントが動作しているＯＳ以外のＯＳに障害が生じたか否かの情報を獲得する（ステップ４００）。

【００２２】

（２）各監視エージェント２０５、２０６は、自監視エージェントが動作しているＯＳ以外のＯＳであって、運用系のＯＳに障害が生じたか否かを判定し、障害が生じていなければ、何の処理も行わず、ここでの処理を終了する（ステップ４０１）。

【００２３】

（３）運用系のＯＳであるＯＳ－１に障害を生じたとすると、ステップ４０１での判定で、待機系ＯＳで動作している監視エージェント２０６が運用系のＯＳであるＯＳ－１に障害を生じたことを検出することになる。運用系のＯＳであるＯＳ－１に障害を生じたことを検出すると、待機系ＯＳ－２の監視エージェント２０６は、計算機資源割り当てテーブル３１０より、障害が起ったとき計算機資源をどのように変更するかを獲得する（ステップ４０２）。

【００２４】

（４）ここで説明している例は、待機系ＯＳで動作している監視エージェント２０６が運用系ＯＳでの障害を検出したとしているので、運用系ＯＳ－１で障害が生じたたとき、待機系ＯＳは、運用系の処理を引き継ぎ、引き継いだ後、運用系ＯＳとして動作することになる。このため、計算機資源として、今まで待機系ＯＳ用のＣＰＵの割り当て率であったものを、運用系の値に変更することになり、監視エージェント２０６は、この計算機資源の変更を実際にＣＰＵの動作を管理するＯＳ管理制御部２００に通知する（ステップ４０３）。

【００２５】

（５）そして、ＯＳ管理制御部が実際の処理を行った後、ＯＳ状態テーブル３２０を変更する。ここでの変更は、ＯＳ状態テーブル３２０の動作状態と、系の種類と、ＣＰＵ利用率との変更であり、障害発生後、今までの運用系ＯＳが待機系

ＯＳに、今までの待機系ＯＳが運用系ＯＳとなり、ＣＰＵ利用率も、この系の変更に従ったテーブルに設定されている値に変更される。主記憶の利用率についても同様に変更される（ステップ４０４）。

【００２６】

（６）そして、障害が生じて待機系とされたＯＳが復帰した場合に備えて、復帰したＯＳで再び稼動する監視エージェントに対して、今現在の情報、すなわち、復帰したＯＳが待機系として動作することを通達しておき処理を終了する（ステップ４０５）。

【００２７】

図４により説明した例は、計算機状態テーブル３００を監視エージェントが監視するとたが、計算機状態テーブル３００の監視を監視エージェントはでなく、管理端末２３０で管理することもできる。この場合、各ＯＳの状態を集中的に管理端末２３０により管理することができるので、ステップ４０５の処理を行う必要がなくなる。また、管理端末２３０に、クラスタサービスのクライアントサービスに接続して、各クラスタの状況を把握することも可能である。この場合、管理端末２３０は、ステップ４００の処理で、定期的に各ＯＳの稼動状況をクラスタサービスに問い合わせればよい。また、前述において、待機系ＯＳと運用系ＯＳとは、基本的には同一のものが使用されるが、同一のアプリケーションの処理を実行することができるものであれば、異なる種類のＯＳであってもよい。

【００２８】

前述したように、本発明の第１の実施形態は、ＣＰＵの資源を、運用系として稼動するＯＳに多くの利用率で割り当てておき、運用系のＯＳに障害が発生したとき、その処理を引き継いで新たに運用系となるＯＳに、ＣＰＵの資源を多くの利用率で再割り当てすることができ、これにより、計算機資源の有効活用を実現することが可能である。

【００２９】

前述した本発明の第１の実施形態は、ＯＳの運用形態に従って、すなわち、ＯＳが運用系であるか待機系であるかにより計算機資源の割り当て比率を決めて、運用形態が変更になったとき、それに従って計算機資源の割り当てを変更すると

して説明したが、本発明は、各ＯＳ上の処理の負荷の大きさや特定の時間に、計算機資源の割り当てを変更して、計算機資源の効率的な運用を図るようにすることもできる。以下、本発明の第２の実施形態を説明する。

【００３０】

図５は本発明の第２の実施形態に必要な各種のテーブルの構成を説明する図、図６はＯＳに負荷がかかった場合の計算機資源割合変更対策を設定した計算機資源変更テーブルの構成を説明する図、図７は計算機資源の負荷を表す計算機資源変更状態テーブルの構成を説明する図、図８は各ＯＳが計算機資源の変更を行う処理手順について説明するフローチャートである。

【００３１】

図５に示す本発明の第２の実施形態に必要な各種のテーブルは、計算機システム全体の資源と、それに対する各ＯＳの資源利用設定とを表すもので、主記憶装置のデータ領域５００に格納されている。このデータ領域５００には、ＣＰＵ使用テーブル５１０、主記憶使用テーブル５２０、外部装置使用テーブル５３０、物理ＣＰＵ設定テーブル５４０、ＯＳ設定テーブル５５０が格納されている。

【００３２】

ＣＰＵ使用テーブル５１０には、項目５１１として、稼動しているＣＰＵ数と稼動していないＣＰＵ数とが登録され、これらに対するＣＰＵ数５１２の値が登録されており、資源変更のときに、これらのデータが用いられる。主記憶使用テーブル５２０には、項目５２１として、使用されているメモリ容量と使用されていないメモリ容量とが登録され、これらに対する値５２２が登録されている。また、外部装置使用テーブル５３０には、項目５３１として、各外部装置であるＩ／Ｏが登録され、それらが使用されているか否かを示す情報５３２が登録されている。物理ＣＰＵ設定テーブル５４０は、ＣＰＵ群１１０の中で、独立して用いるＣＰＵの設定状況を示している。このテーブル５４０には、項目５４１として、分割のための各ＯＳのＩＤが登録され、それらの各ＯＳに対して、それらのＩＤを持つＯＳ相互間での優先度５４２、初期に分割して割り当てたＣＰＵ数である初期個数５４３、このテーブルを定期的に更新する監視エージェントにより実際に計算機資源を変更したときに更新された現在のＣＰＵ数を表す現在個数５４

４が登録されている。優先度５４２は、ＯＳがさらに多くのＣＰＵを要求した場合に、その変更を行う場合の優先度である。ＯＳ設定テーブル５５０は、各ＯＳの初期設定を登録するテーブルであり、このテーブル５５０には、各ＯＳのＯＳ名５５１に対して、各ＯＳ間の実行優先度５５２、ＯＳの実行系の種類５５３、物理ＣＰＵ設定テーブルのＩＤを表すＩＤ５５４、ＣＰＵサービス比５５５、主記憶の使用容量５５６、使用外部装置５５６、５５７がそれぞれ登録されている。

【００３３】

前述において、使用外部装置は、計算機システム１００の接続状況によりその個数が変更される。また、ＯＳの実行系の種類５５３とは、継続的に処理を行うトランザクション系と、蓄積してある処理を集中して行うバッチ系とである。本発明の第２の実施形態は、この処理系の性質の違いを考慮して、計算機資源を変更することにより、無駄のない計算機資源の割り当ての制御を行う。

【００３４】

図６に示すＯＳに負荷がかかった場合の計算機資源割合変更対策を設定した計算機資源変更テーブル６００には、ＯＳ名６０１と、そのＯＳに対応する各変更ルールのＩＤ６０２とが登録され、さらに、これらのそれぞれに対して次に説明するような情報が登録される。待機フラグ６０３は、各ルールを示すＩＤ６０２の条件に対して、実際に計算機資源の変更が可能でない状況がある場合、計算機資源の変更を再度行うか否かを示すフラグであり、この待機フラグ６０３には数値が入れられる。このフラグ６０３は、定期的に監視を行うエージェントが、後述する図８のフローによる処理を実行するたびに、計算機資源の変更が可能であるか否かを調査するために使用される。その調査回数が待機フラグの値である。ＯＳ名６０１に対して、監視エージェント２０５、２０６は、ＯＳの負荷を、計算機資源変更条件６０４に基づいて調査する。計算機資源変更条件６０４には、各計算機資源のＣＰＵ６０６、主記憶６０７、Ｉ／Ｏ（外部装置）６０８に関する閾値データが格納されている。そして、監視エージェント２０５、２０６は、調査の結果、登録条件を満たせば、計算機資源変更設定６０５の中の各計算機資源であるＣＰＵ６０９、主記憶６１０、Ｉ／Ｏ６１１の値に基づいて設定要求を

行う。また、ＯＳ共通の設定がある場合、各ＯＳがＯＳ共通のルールを選択する。

【００３５】

図７に示す各ＯＳの計算機資源の負荷を表す計算機資源変更状態テーブル７００は、ＯＳ名７０１と各ＯＳの状態と格納しており、各ＯＳの状態として、各ＯＳに割り当てられている計算機資源の負荷が格納され、監視エージェントは、この負荷の値を考慮して計算機資源変更の判定基準とする。また、変更後の資源状態も管理し、今後値を戻すためのデフォルト値と共に管理される。各ＯＳ名７０１に対する現在の算機資源として、ＣＰＵ７０２、メモリ７０４、Ｉ／Ｏ７０６、Ｉ／Ｏ２（７０８）が登録され、そのそれぞれについての負荷である、ＣＰＵ負荷７０３、メモリ負荷７０５、Ｉ／Ｏ負荷７０７、Ｉ／Ｏ２負荷７０９が格納される。この計算機資源の状態は、計算機資源変更ルールを用いるか否かの判断基準となる。

【００３６】

次に、図８に示すフローを参照して、各ＯＳが計算機資源の変更を行う処理手順について説明する。この処理は、監視エージェントがＯＳと連携して実行する処理である。

【００３７】

（１）監視エージェントは、定期的に各ＯＳと連携してＯＳの稼動状態を調査し、その状態をＯＳ状態テーブルである計算機資源の負荷を表す計算機資源変更状態テーブル７００に格納し、計算機資源変更テーブル６００の情報に基づいて、連携しているＯＳの負荷が高負荷となっているか否かをチェックする（ステップ８００、８０１）。

【００３８】

（２）ステップ８０１でのチェックで、そのＯＳの負荷が高負荷となっていなかった場合、ＯＳに割り当てられている資源がデフォルト値と同一か否かをチェックし、デフォルト値と同一であった場合、計算機資源を元の状態に戻す必要がないので、何も行わずに処理を終了する（ステップ８０５）。

【００３９】

（３）ステップ８０５のチェックで、ＯＳに割り当てられている資源がデフォルト値と同一でなかった場合、資源の状態を以前の状態に戻すことができるか否かをチェックし、資源の状態を以前の状態に戻すことができる状態ではない場合、何も行わずに処理を終了する（ステップ８０６）。

【００４０】

（４）ステップ８０１のチェックで、そのＯＳの負荷が高負荷となっていた場合、そのＯＳの優先度が１番であるか否かをＯＳ設定テーブル５５０により調べ、優先度が１番でない場合、計算機資源変更テーブル７００から他のＯＳの負荷状況を獲得し、他のＯＳの負荷が大きくなく、現在計算機資源変更が可能か否かを調べる（ステップ８０２、８０３）。

【００４１】

（５）ステップ８０２のチェックで、そのＯＳの優先度が１番であった場合、または、ステップ８０３のチェックで、他のＯＳの負荷が大きくなく、現在計算機資源の変更が可能であると判定された場合、あるいは、ステップ８０６のチェックで、資源の状態を以前の状態に戻すことができる状態であると判定した場合、計算機資源変更テーブル６００から資源変更のデータを獲得する（ステップ８０７）。

【００４２】

（６）ステップ８０３のチェックで、他のＯＳの負荷が大きく、現在計算機資源の変更を行うことが不可能であると判定した場合、連携しているＯＳ及び他のＯＳの待機フラグ６０３が設定されているか否かをチェックし、いずれかのＯＳの待機フラグが設定されていれば、連携しているＯＳの待機フラグの設定を変更（＋１）して、ここでの処理を終了する（ステップ８０４）。

【００４３】

（７）ステップ８０４のチェックで、待機フラグ６０３が設定されていなかった場合、計算機資源変更テーブル６００から資源変更のデータを獲得し、それに見合う計算機資源の追加資源量を獲得する（ステップ８０８）。

【００４４】

（８）前述したステップ８０７あるいはステップ８０８の処理後、実際に計算機

資源を変更可能なＯＳ連携制御部に、計算機資源の変更要求を出し、計算機資源の変更処理の終了後、状態テーブル７００を変更する（ステップ８０９、８１０）。

【００４５】

本発明の第２の実施形態によれば、前述したような処理を行うことによって、ＯＳの負荷の状態を監視し、その負荷の状態に従って、計算機資源の変更要求を行って、効率的に資源の運用を行うことが可能に、ＯＳに対して計算機資源の割り当てを行うことができる。

【００４６】

前述した本発明の第２の実施形態は、ＯＳの負荷の状態に応じて、ＯＳに割り当てる計算機資源を動的に変更するものであったが、本発明は、複数のＯＳのそれぞれが扱う処理群を管理し、この処理を他のＯＳの処理と比較して優先的に処理を行う必要がある場合、優先的に処理を行う必要があるＯＳに計算機資源を多く割り当てるようにすることもできる。また、本発明は、複数のＯＳのいずれにも割り当てられていない計算機資源を管理しておき、各ＯＳに割り当てられている計算機資源の使用率を監視して、この使用率が高くなったＯＳに、割り当てられていない計算機資源を割り当てるようにすることができる。

【００４７】

また、前述した本発明の第２の実施形態は、バッチ処理系に対する計算機資源の割り当てに適用することにより、集中的なバッチ処理を行うことが可能となり、バッチ処理を効率的に実行することができる。以下、この例について説明する。

【００４８】

図９はバッチ処理系に対する計算機資源変更用のテーブルの構成を説明する図、図１０はバッチ処理系の計算機資源変更の処理動作を説明するフローチャートである。ここで説明する例は、バッチ処理系の計算機資源変更を、そのバッチ処理が行われる毎に、他の計算機資源に余裕があり余っていれば、その計算機資源をバッチ処理系割り当てて、そのバッチ処理を集中的に行わせることを可能にしたものである。

【００４９】

図９に示す計算機資源変更テーブル９００には、バッチ処理のジョブ名９０１に関して、ジョブを実行するＯＳ名９０２、ジョブの開始時間９０３、終了時間９０４、及び、開始、終了の時間の間での各計算機資源の変更条件９０６が計算機資源変更テーブル６００の場合と同様に登録される。また、計算機資源変更テーブル６００の場合と同様な待機フラグ９０５も設定されている。

【００５０】

次に、図１０に示すフローを参照して、バッチ処理系の計算機資源変更の処理動作について説明する。

【００５１】

（１）監視エージェント２０５、２０６は、テーブル９００によりジョブの開始時間を監視し、現時間がジョブの開始時間になったか否かを調べ、ジョブ開始時間でなければ、何の処理も行わずに、ここでの処理を終了する（ステップ１０００、１００１）。

【００５２】

（２）ステップ１００１のチェックで、現時間がジョブの開始時間であれば、図８により説明した場合と同様に、バッチ系のＯＳの優先度が１番となっているか否かを調べ、優先度が１番となっている場合、それなりの計算機資源が割り当てられているので、何の処理も行わずに、ここでの処理を終了する（ステップ１００２）。

【００５３】

（３）ステップ１００２のチェックで、優先度が１番でない場合、他のＯＳの負荷を調べ、他のＯＳに大きな負荷が掛かっている場合、計算機資源の割り当て不可能として、何の処理も行わずに、ここでの処理を終了する（ステップ１００３）。

【００５４】

（４）ステップ１００３のチェックで、他のＯＳに負荷が掛かっていなかった場合、待機フラグを調べて他に計算機資源変更を要求して待ちになっているＯＳがあるか否かを調べて、待機フラグが設定されているＯＳがあれば、待機フラグ９

０３を設定して処理を終了する（ステップ１００４）。

【００５５】

（５）ステップ１００４のチェックで、待機フラグが設定されていなかった場合、計算機資源の変更が可能であると判定し、計算機資源変更テーブル９００から、変更条件を獲得し、それを実際に計算機資源を変更するＯＳ管理制御部２４０に通達する。そして、資源の変更が終了すれば、変更後ＯＳ状態テーブルに状況を書き込んで処理終了する（ステップ１００５～１００７）。

【００５６】

前述した処理動作の後、バッチ処理系は、変更された計算機資源を使用してバッチ処理を実行する。そして、監視エージェント２０５、２０６は、その処理の終了後、または、計算機資源変更テーブル９００のジョブ終了時間９０４になった場合、計算機資源の状態を変更前の元の状況に戻す。

【００５７】

前述したように、計算機の状態を監視エージェントにより管理し、適宜計算機資源を変更することにより、計算機資源を複数のＯＳに対して有効に割り当てることが可能である。

【００５８】

一般に、マルチプロセッサ環境の計算機をサーバ機として用いる場合、複数のユーザがサーバ計算機にアクセスする。その際、アクセスする計算機の処理能力を保証するため、計算機資源が分割してユーザ毎に割り当てられる。このようにサーバ計算機に対して本発明による計算機資源を変更する手段を与え、その変更の状況に応じて課金を行うようにすることが可能である。この課金方法によって、ユーザに柔軟なサーバ機の利用を行わせることができる。以下、計算機資源変更技術を用いた課金方法を本発明の第３の実施形態として説明する。

【００５９】

図１１は本発明の第３の実施形態に必要なユーザ課金テーブルとユーザ登録テーブルとを持つユーザ設定領域の構成を説明する図、図１２は課金計算に必要な各種の課金基準テーブルを持つ計算機資源課金基準領域の構成を説明する図、図１３は計算機資源の利用を基準とした課金処理の手順を説明するフローチャート

である。

【００６０】

図１１に示すユーザ設定領域１１００は、主記憶装置ないに格納され、ユーザ課金テーブル１１１０とユーザ登録テーブル１１２０とにより構成される。ユーザ課金テーブル１１１０は、計算機資源を変更するユーザからの要求に応じて、そのテーブルの内容が更新されていく。ユーザ課金テーブル１１１０には、各ユーザを表すユーザＩＤ１１１１に対して、使用加算時間１１１２、ＣＰＵ課金状況１１１３、１１１４、主記憶課金状況１１１５、１１１６、外部装置（Ｉ／Ｏ）課金状況１１７、１１１８、合計課金状況１１１９に各課金値が更新可能に設定される。ユーザ登録テーブル１１２０には、ユーザが初期に登録したデータが格納されており、各ユーザを表すユーザＩＤ１１２１に対して、このテーブルの各項目である使用開始時間１１２２、使用終了時間１１２３、ＣＰＵ登録１１２４、主記憶登録１１２５、外部装置（Ｉ／Ｏ）登録１１２６が初期設定として登録され、また、それに対する標準料金１１２７も計算されて登録されている。

【００６１】

本発明の第３の実施形態は、前述のテーブル１１１０、１１２０への設定情報を基準として課金を行っていく。前述のテーブルにおいて、テーブル内の丸印は、課金がかからないことを表している。すなわち、ユーザ１がＣＰＵを１個使用しても、標準料金以外に料金がかからないことを意味する。そして、この基準以上の計算機資源をユーザが使用した場合、それに対する課金が行われ基準料金に加算されることになる。

【００６２】

図１２に示す計算機資源課金基準領域１２００には、それぞれの条件に基づいて課金設定を行い、課金を行うためのテーブルであるＣＰＵ課金基準テーブル１２１０、メモリ課金基準テーブル１２３０、外部装置課金基準テーブル１２５０が設けられており、これらのテーブルには課金の計算の基準となる値が格納されている。図１２における例において、各テーブルの項目は、各計算機資源の使用時間と使用量との関係で決められる課金基準である。

【００６３】

次に、図１３に示すフローを参照して、計算機資源の利用を基準とした課金手順について説明する。

【００６４】

（１）まず、ユーザの使用確認を行い、そのユーザの条件である使用ユーザ指定処理の実行要求を獲得する。その後、計算機資源の要求があるか否かを判定し、もしなければ、資源の変更を行わず、他のユーザ等からの要求の処理に移る（ステップ１３００～１３０２）。

【００６５】

（２）ステップ１３０２の判定で、ユーザが資源変更の要求を行っていれば、その条件に基づいてＯＳ管理制御部２００に処理の要求を行い、計算機資源の変更が行われた後、計算機資源状態テーブル７００を変更し、課金テーブルにその条件を登録する（ステップ１３０３、１３０４）。

【００６６】

（３）そして、その課金状態が終了すると、計算機資源の状態を元の状態に戻し、課金テーブルに、その使用時間と課金情報とを格納して処理を終了する（ステップ１３０５）。

【００６７】

前述したような本発明の第３の実施形態によれば、ユーザは、柔軟に計算機環境を変更することができ、必要のあるときのみ、機能の高い計算機システムの提供を受け、必要がない場合、最低限の機能の計算機システムの提供を受けて、コスト的に柔軟に対応することが可能となる。

【００６８】

図１４は本発明の第４の実施形態による計算機システム上で動作する各ＯＳをクラスタ構成とした場合の概念的な構成を示すブロック図、図１５は共有メモリ通信の送信処理手順を説明するフローチャート、図１６は共有メモリ通信の受信処理手順を説明するフローチャートであり、以下、本発明の第４の実施形態について説明する。

【００６９】

本発明の第４の実施形態は、ＯＳ管理制御部２００に各ＯＳに共通の共有メモ

リ１４００をおき、その共有メモリに通信用のフォーマットを作成して書き込み読み出しを行うことにより、通信ケーブルを必要としない擬似的な通信網を作成した共有メモリ通信を行うものである。そして、この実施形態は、図１４に示すように、図２に示した構成において、ＯＳ管理制御部にＯＳ間の共有メモリ１４００を付属させたものであり、クラスタ構成を組むときの内部通信網として共有メモリ通信をおこなうことも可能である。また、この実施形態は、図２に示した例で説明したケーブル２１２を不要として、独自の内部通信網を確立することができる。また、共有メモリ通信は、各ＯＳ領域に設けられる共有メモリ通信ドライバ１４０１、１４０２により行われる。

【００７０】

共有メモリ通信の送信手順は、図１５に示すフローに従って、次のように行われる。

【００７１】

（１）共有メモリ通信ドライバは、ＯＳ上のプログラムから送信要求を受け取ると、送信データを通信用のフォーマットに変換する。例えば、通信として、ＴＣＰ／ＩＰによるイーサネットの通信を想定しているのであれば、そのフォーマットを生成する（ステップ１５００、１５０１）。

【００７２】

（２）そして、共有メモリ１４００への書き込み要求をＯＳ管理制御部２００に要求する（ステップ１５０２）。

【００７３】

（３）ＯＳ管理制御部２００は、共有メモリ１４００に空きがあるか否かを判定し、空きがなければ、一度処理を終わり、共有メモリ１４００に空きができるまで待つ（ステップ１５０３）。

【００７４】

（４）ステップ１５０３判定で、共有メモリ１４００に空きがあれば、送信データを共有メモリに書き込む（ステップ１５０４）。

【００７５】

前述した処理において、実際に共有メモリに送信データを書き込むのはＯＳ管

理制御部２００であるので、通信ドライバ１４０１、１４０２は、ＯＳ管理制御部に対して書き込みデータを伝えることになる。

【００７６】

共有メモリ通信の受信手順は、図１６に示すフローに従って、次のように行われる。なお、ＯＳは、一定時間毎に共有メモリ通信ドライバに受信要求を行い、受信すべきデータがある場合に、受信データを受け取るようにすることができ、あるいは、送信側となったＯＳの共有メモリ通信ドライバが、受信側に何らかの方法でデータの送信を知らせるようにすることもできる。

【００７７】

（１）共有メモリ通信ドライバは、ＯＳ上のプログラムから受信要求を受け取ると、共有メモリ１４００に受信用のデータがあるか否かをチェックする。すなわち、共有メモリ１４００に自分が受信するデータが書き込まれているか否かを調べる。そして、もし受信データがなければ、ここでの処理を終了する（ステップ１６００、１６０１）。

【００７８】

（２）ステップ１６０１のチェックで、共有メモリに受信データがあれば、その受信データを共有メモリから読み出し、受信データの通信フォーマットを解析して、受信データを取り出す（ステップ１６０２、１６０３）。

【００７９】

前述した例によれば、共有メモリを用いた送受信方法により、通信用のケーブルを備えることなく、共通するメモリエリアを用いて擬似的に通信ケーブルを実現することが可能である。この方法は、クラスタにおけるネットワーク通信等にも利用することができる。また、メモリ内に不揮発的に扱えるメモリエリアあれば、そこを擬似的な外部記憶装置にみたてて、その間を、ＳＣＳＩインターフェイスを用いてアクセスすることにより、擬似的なＳＣＳＩ通信を行うことも可能である。

【００８０】

【発明の効果】

以上説明したように本発明によれば、１台の計算機に複数のＯＳを稼動させた

計算機システムであって、各ＯＳに対する計算機資源の変更、再配置を行うことが可能な環境の中で、クラスタシステムを構成することにより、各ＯＳの運用状況を監視して、運用系ＯＳの障害時に計算機資源を別の正常なＯＳに多く割り当てて、それを運用系とすることにより、障害に関わらず、計算機システムの処理能力を変えることなく計算機システムを稼働させることができる。

【００８１】

また、本発明によれば、ＯＳの負荷を監視して、その負荷に応じて、計算機資源を割り当てることが可能であるので、ＯＳ間の処理能力を必要に応じて調整することが可能である。

【図面の簡単な説明】

【図１】

本発明の第１の実施形態による計算機システムの構成を示すブロック図である。

【図２】

計算機システム上で動作する各ＯＳをクラスタ構成とした場合の概念的な構成を示すブロック図である。

【図３】

計算機の状態を管理するテーブルの構成を説明する図である。

【図４】

ＯＳの障害時の計算機資源の変更処理の動作を説明するフローチャートである。

【図５】

本発明の第２の実施形態に必要な各種のテーブルの構成を説明する図である。

【図６】

ＯＳに負荷がかかった場合の計算機資源割合変更対策を設定した計算機資源変更テーブルの構成を説明する図である。

【図７】

計算機資源の負荷を表す計算機資源変更状態テーブルの構成を説明する図である。

【図８】

各ＯＳが計算機資源の変更を行う処理手順について説明するフローチャートである。

【図９】

バッチ処理系に対する計算機資源変更用のテーブルの構成を説明する図である。

【図１０】

バッチ処理系の計算機資源変更の処理動作を説明するフローチャートである。

【図１１】

本発明の第３の実施形態に必要なユーザ課金テーブルとユーザ登録テーブルとを持つユーザ設定領域の構成を説明する図である。

【図１２】

課金計算に必要な各種の課金基準テーブルを持つ計算機資源課金基準領域の構成を説明する図である。

【図１３】

計算機資源の利用を基準とした課金処理の手順を説明するフローチャートである。

【図１４】

本発明の第４の実施形態による計算機システム上で動作する各ＯＳをクラスタ構成とした場合の概念的な構成を示すブロック図である。

【図１５】

共有メモリ通信の送信処理手順を説明するフローチャートである。

【図１６】

共有メモリ通信の受信処理手順を説明するフローチャートである。

【符合の説明】

１００　計算機システム

１１０　プロセッサ

１２０　主記憶装置

１３０　外部入出力装置群

１４０　複数ＯＳ管理制御支援装置

１５０　バス

２００　複数ＯＳ管理制御部

２０１、２０２　第１、第２ＯＳ

２０３、２０４　クラスタサービス

２０５、２０６　監視エージェント

２０７、２０８　アプリケーション

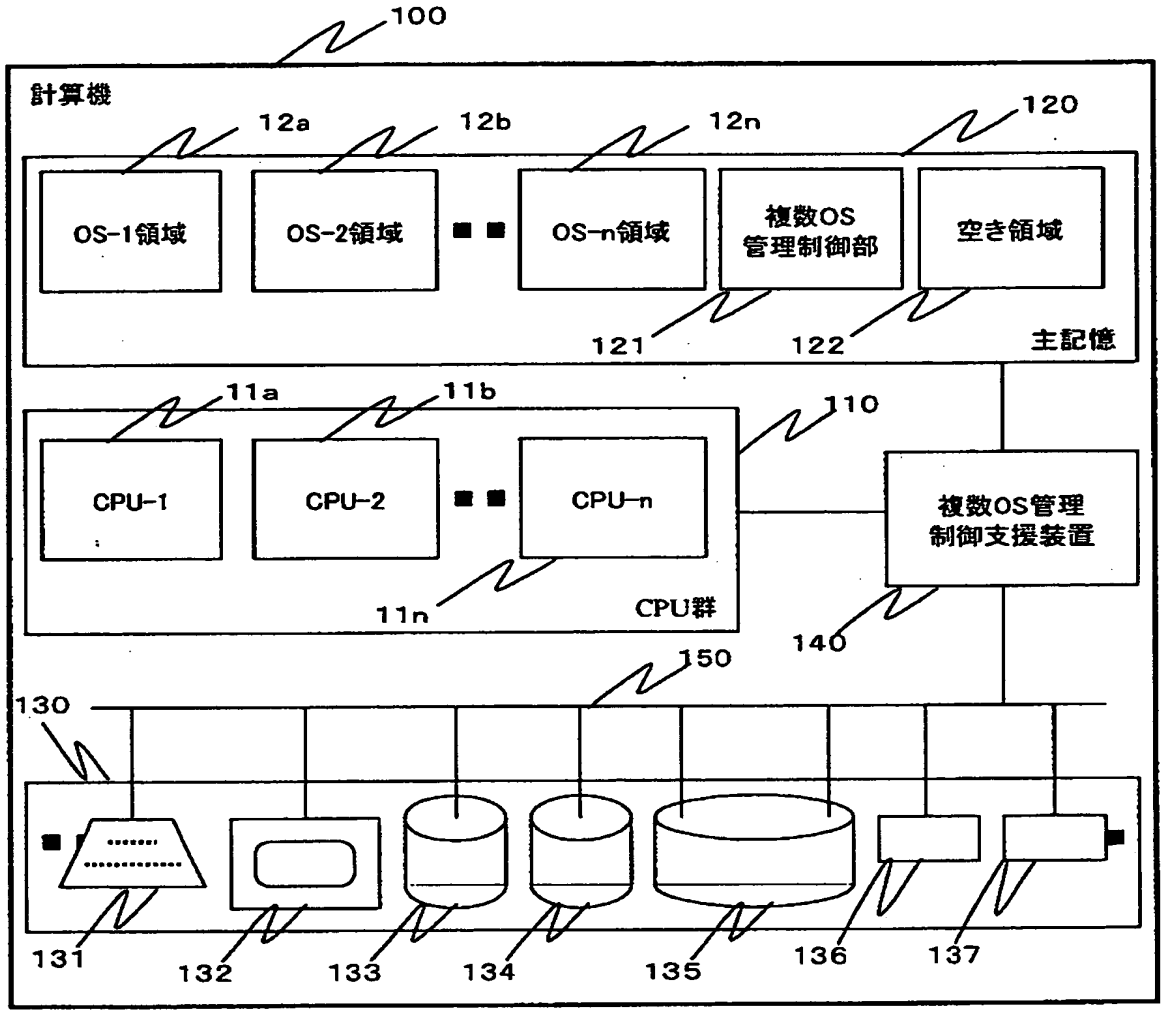
２２１、２２２　第１、第２ＯＳ用ＣＰＵ群

２３０～２３２　端末

【書類名】　　図面

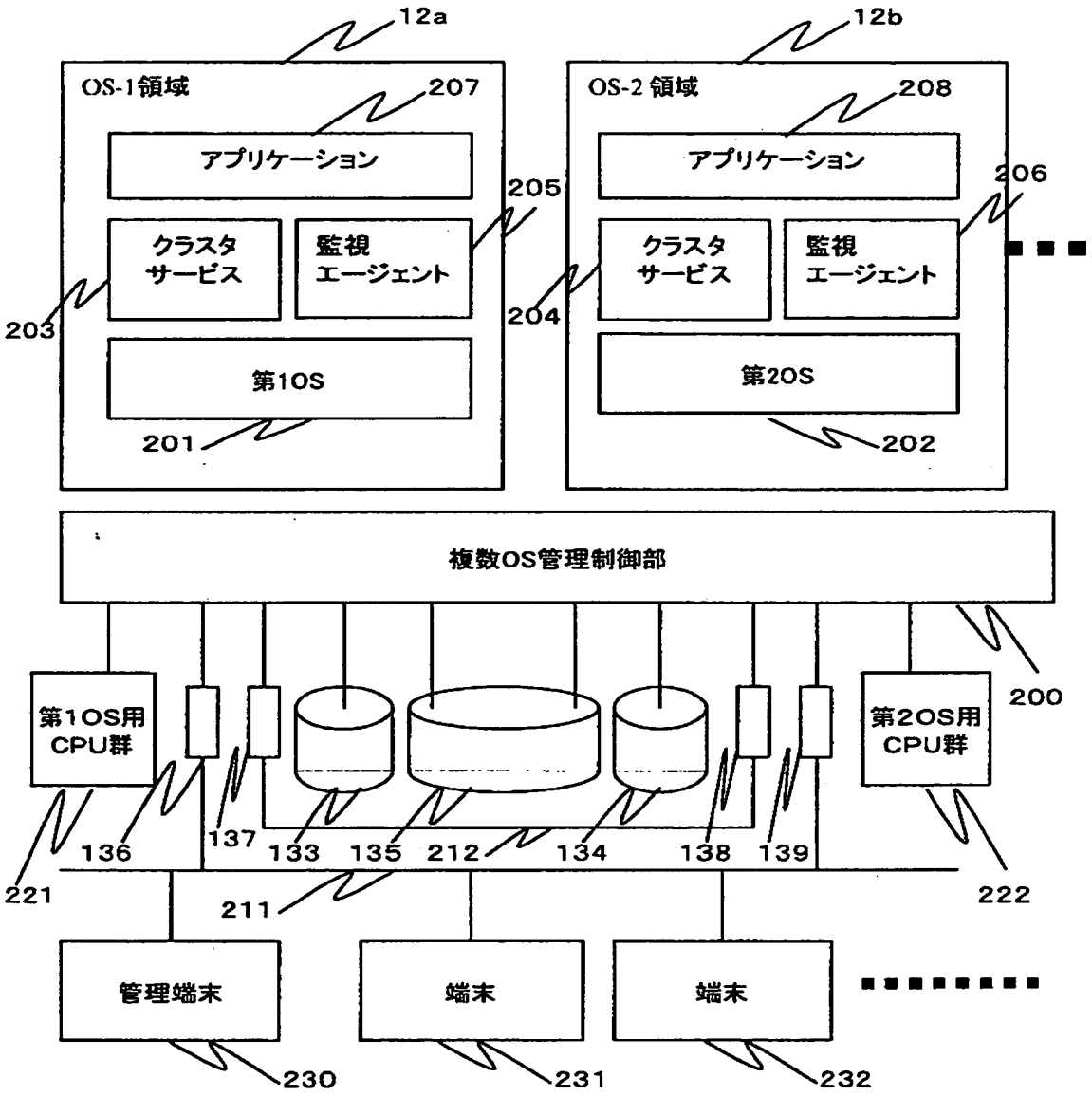
【図１】

図1


100
計算機
12a
12b
12n
120
OS-1領域
OS-2領域
OS-n領域
複数OS
管理制御部
空き領域
121
122
主記憶
11a
11b
110
CPU-1
CPU-2
CPU-n
複数OS管理
制御支援装置
11n
CPU群
140
150
130
131
132
133
134
135
136
137

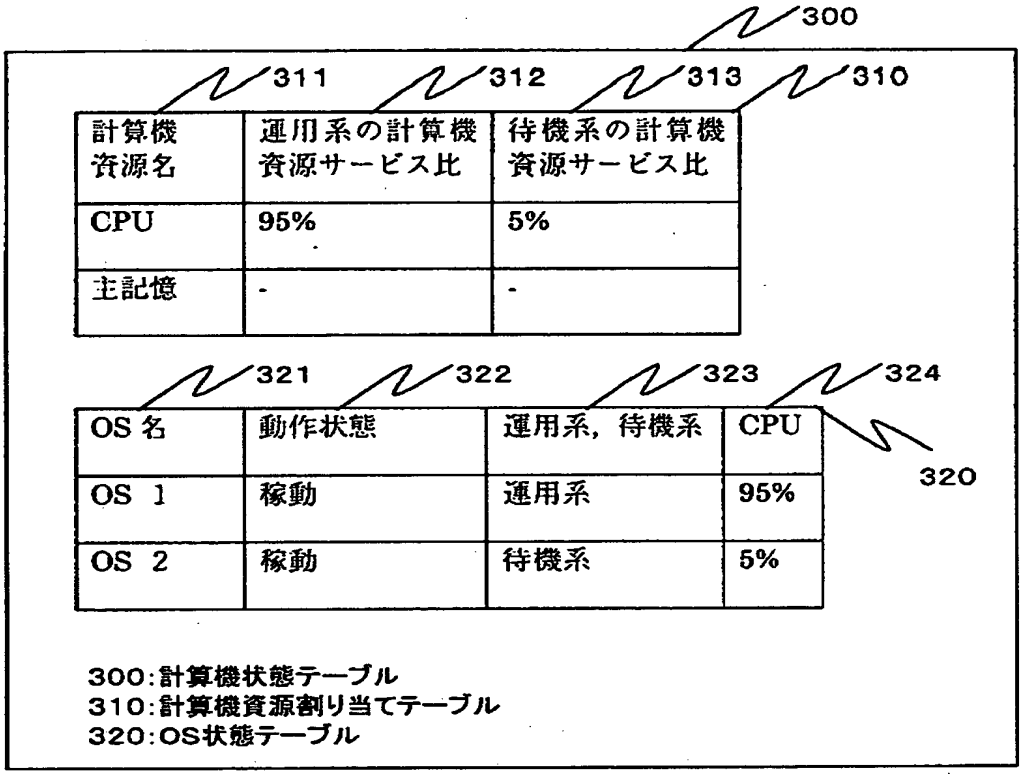
【図２】

図2


12a
12b
OS-1領域
207
OS-2 領域
208
アプリケーション
アプリケーション
205
206
クラスタ
サービス
監視
エージェント
クラスタ
サービス
監視
エージェント
203
204
第1OS
第2OS
201
202
複数OS管理制御部
200
第1OS用
CPU群
第2OS用
CPU群
136
137
133
135
212
134
138
139
221
222
211
管理端末
端末
端末
230
231
232

【図３】

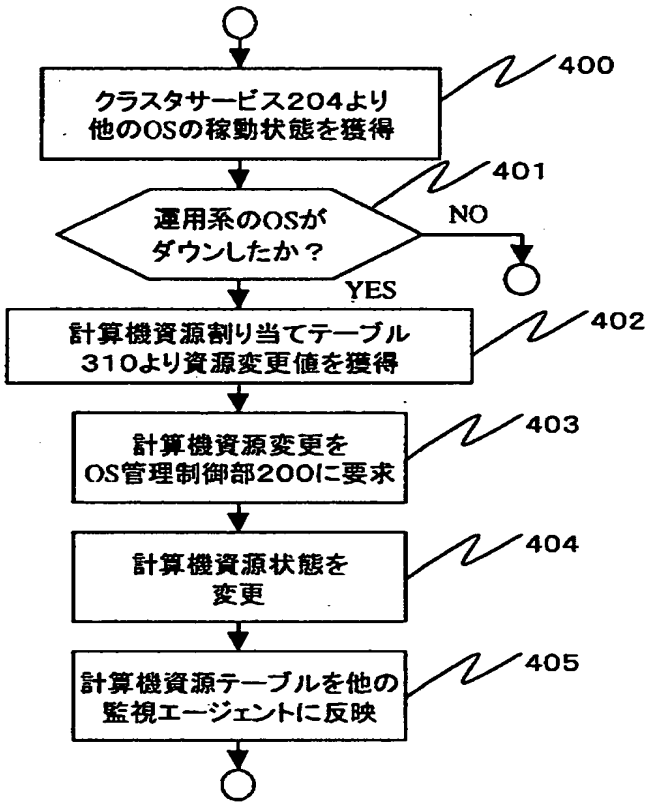
図3



出証特２０００－３１０４４０９

【図４】

図4


400
クラスタサービス204より
他のOSの稼動状態を獲得
401
運用系のOSが
ダウンしたか？
NO
YES
402
計算機資源割り当てテーブル
310より資源変更値を獲得
403
計算機資源変更を
OS管理制御部200に要求
404
計算機資源状態を
変更
405
計算機資源テーブルを他の
監視エージェントに反映

【図５】

図5

500

510 511 512

項目	値
使用 CPU 数	
空き CPU 数	

520 521 522

項目	値
使用メモリ数	
空きメモリ数	

530 531 532

項目	使用／未使用
I/O 1	
I/O 2	
I/O n	

540 541 542 543 544

ID	優先度	初期個数	現在個数
1	1	2	2
2	2	1	1

550 551 552 553 554 555 556 557 558

OS 名	優先度	種類	ID	CPU	主記憶	I/O 1	I/O 2
OS 1							
OS 2							
OS n							

500:計算機資源設定領域
510:CPU使用テーブル
520:主記憶使用テーブル
530:外部装置使用テーブル
540:物理CPU設定テーブル
550:OS設定テーブル

【図６】

図6

OS名 (600)	ＩＤ (601)	待機フラグ (602)	計算機資源変更条件 (603)			計算機資源変更設定 (604)		
			CPU (606)	主記憶 (607)	I/O (608)	CPU (609)	主記憶 (610)	I/O (611)
OS1								
OS2								
Osn								
OS共通								

600:計算機資源変更テーブル1

【図７】

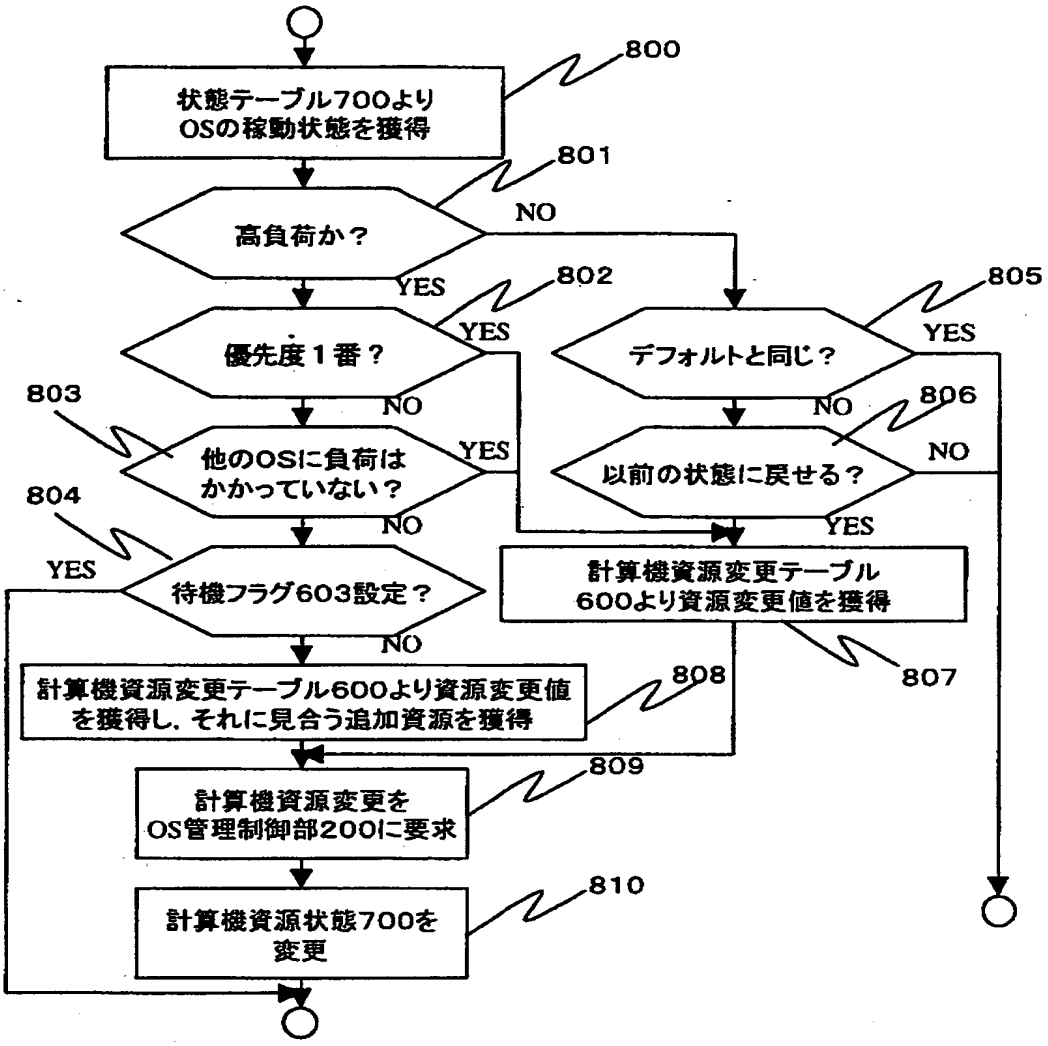
図7

700	701	702	703	704	705	706	707	708	709
OS 名	CPU	CPU 負荷	主記憶	主記憶 負荷	I/O	I/O 負荷	I/O 2	I/O 2 負荷	
OS 1									
OS 2									
OS n									

700:計算機資源変更状態テーブル

【図８】

図8


800
状態テーブル700より
OSの稼動状態を獲得
801
高負荷か？
NO
YES
802
優先度１番？
YES
NO
803
他のOSに負荷は
かかっていない？
YES
NO
804
YES
待機フラグ603設定？
NO
805
デフォルトと同じ？
YES
NO
806
以前の状態に戻せる？
NO
YES
計算機資源変更テーブル
600より資源変更値を獲得
807
計算機資源変更テーブル600より資源変更値
を獲得し、それに見合う追加資源を獲得
808
計算機資源変更を
OS管理制御部200に要求
809
計算機資源状態700を
変更
810

【図９】

図9

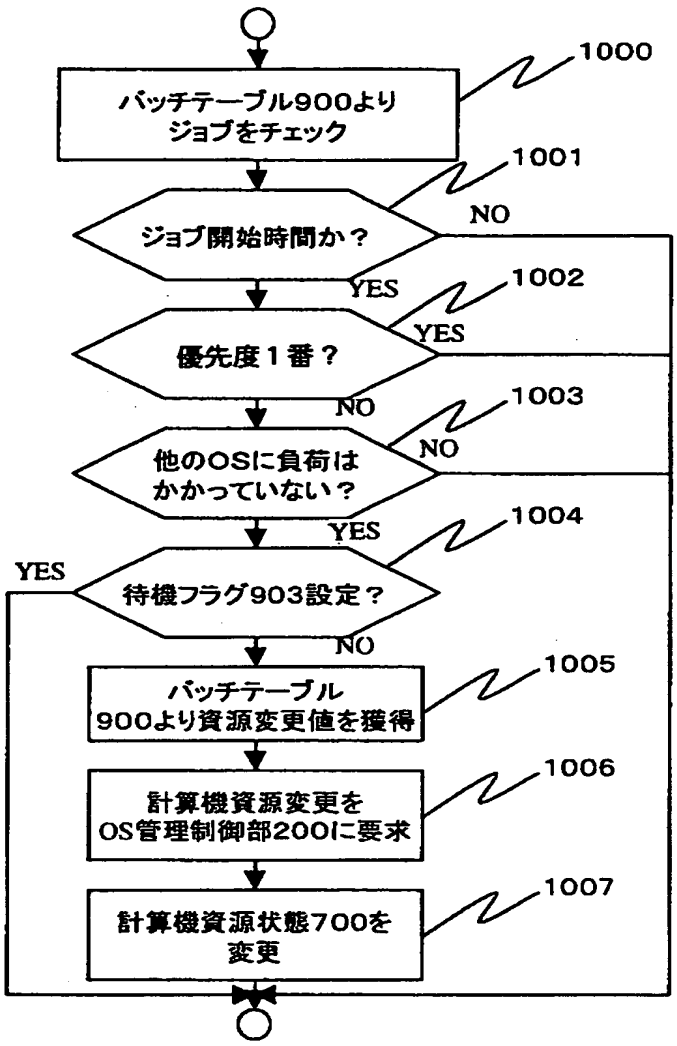
ジョブ名 (900)	OS名 (901)	ジョブ開始時間 (902)	ジョブ終了時間 (903)	待機フラグ (904)	計算機資源要求値 (905)		
					CPU (907)	主記憶 (908)	I/O (909)
1							
2							
n							

(906)

900:計算機資源変更テーブル2

【図１０】

図10


1000
バッチテーブル900より
ジョブをチェック
1001
ジョブ開始時間か？
NO
YES
1002
優先度１番？
YES
NO
1003
他のOSに負荷は
かかっていない？
NO
YES
1004
待機フラグ903設定？
YES
NO
1005
バッチテーブル
900より資源変更値を獲得
1006
計算機資源変更を
OS管理制御部200に要求
1007
計算機資源状態700を
変更

【図１１】

図11

1100

1110 1111 1112 1113 1114 1115 1116 1117 1118 1119

ユーザID	加算使用時間	1 CPU	2 CPU	１００M主記憶	２００M主記憶	I/O 1	I/O 2	課金
1		○		○		○	○	
2		○	○	○	○	○	○	
n								

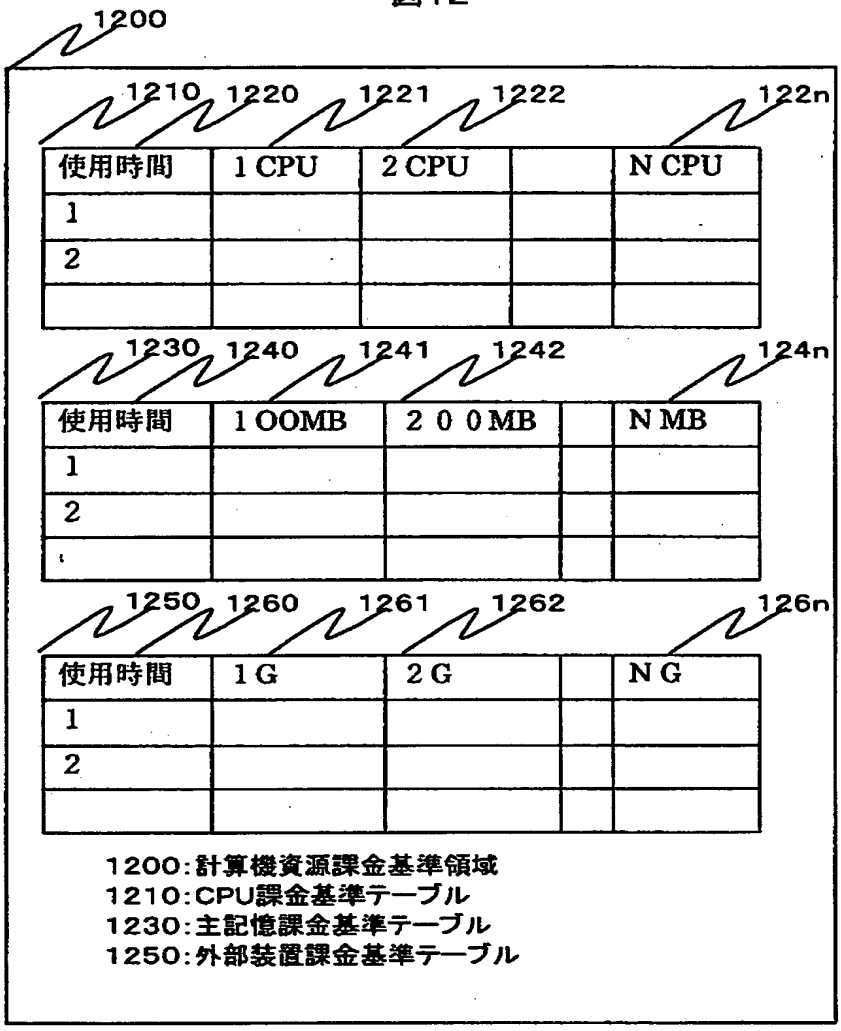
1120 1121 1122 1123 1124 1125 1126 1127

ユーザID	使用開始時間	使用終了時間	CPU	主記憶	I/O	標準料金
1	○	○	1	100	○	
2	8	20	2	200	○	
n						

1100:ユーザ設定領域
1110:ユーザ課金テーブル
1120:ユーザ登録テーブル

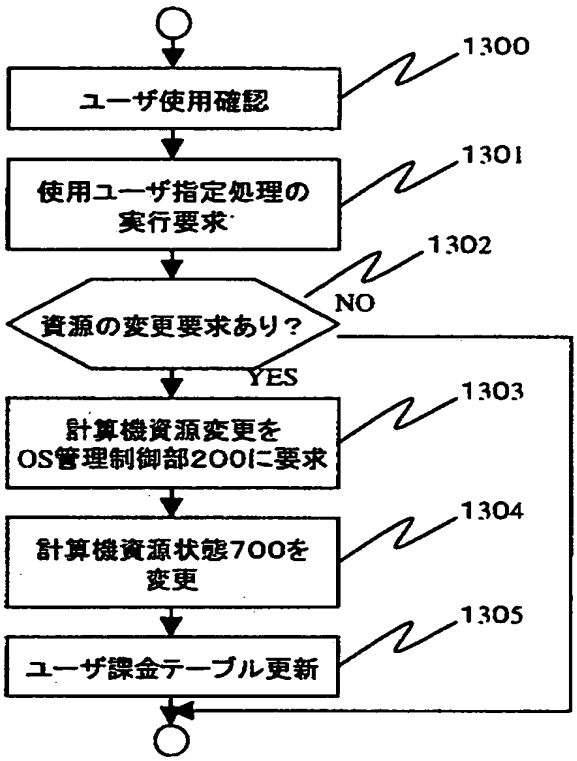
【図１２】

図12


1200
1210
1220
1221
1222
122n
使用時間
1 CPU
2 CPU
N CPU
1
2
1230
1240
1241
1242
124n
使用時間
1 OOMB
２００MB
N MB
1
2
1250
1260
1261
1262
126n
使用時間
1 G
2 G
N G
1
2
1200:計算機資源課金基準領域
1210:CPU課金基準テーブル
1230:主記憶課金基準テーブル
1250:外部装置課金基準テーブル

【図１３】

図13


1300
ユーザ使用確認
1301
使用ユーザ指定処理の
実行要求
1302
資源の変更要求あり？
NO
YES
1303
計算機資源変更を
OS管理制御部200に要求
1304
計算機資源状態700を
変更
1305
ユーザ課金テーブル更新

【図１４】

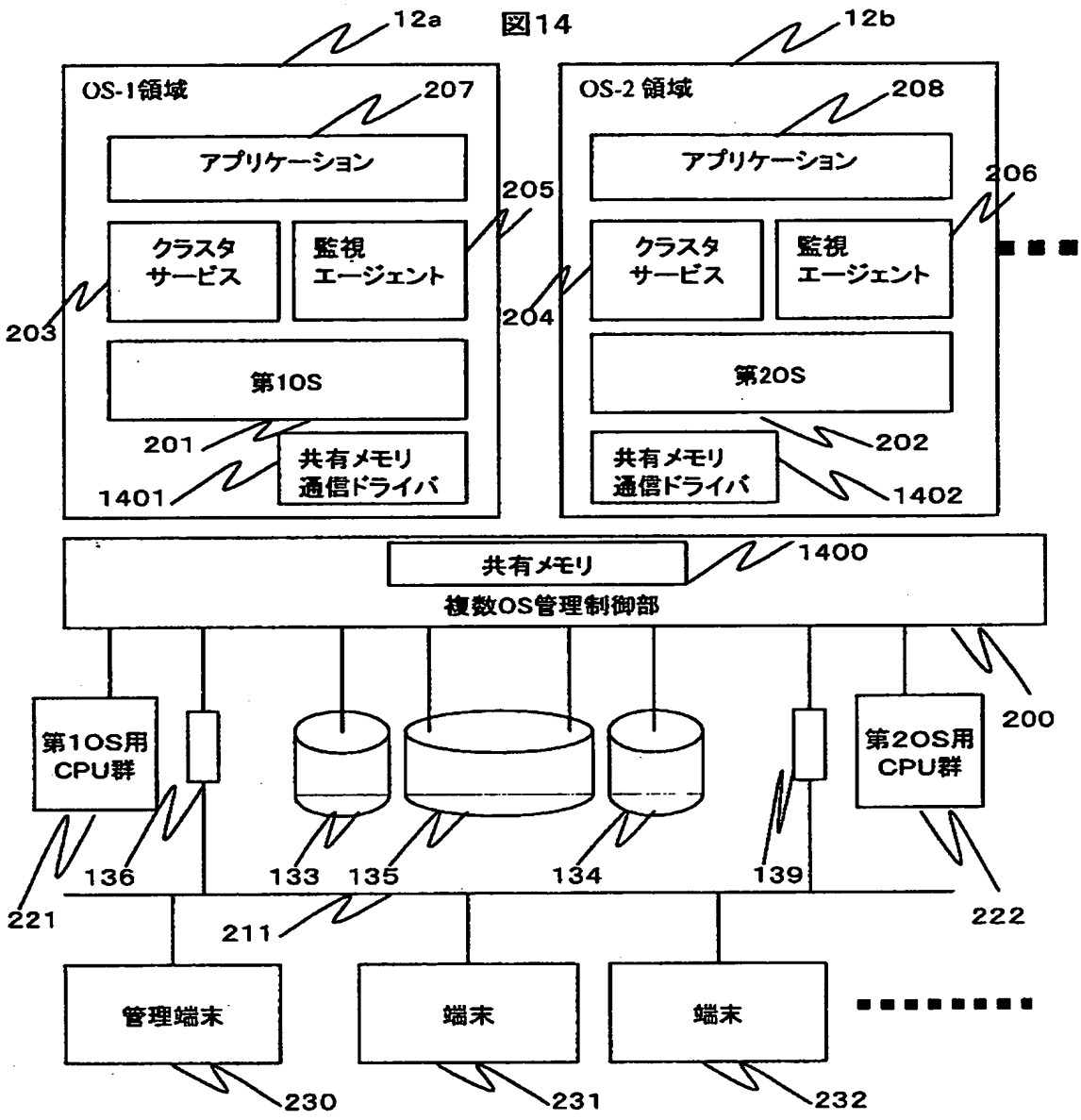
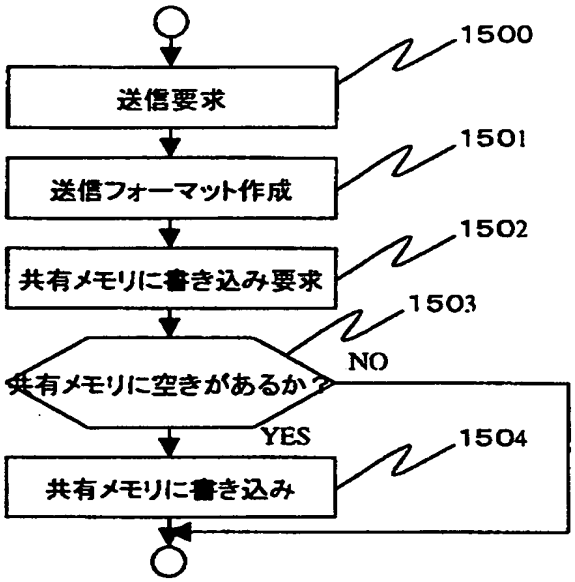


図14
12a
12b
OS-1領域
207
アプリケーション
205
クラスタ
サービス
監視
エージェント
203
第1OS
201
共有メモリ
通信ドライバ
1401
OS-2 領域
208
アプリケーション
206
クラスタ
サービス
監視
エージェント
204
第2OS
202
共有メモリ
通信ドライバ
1402
共有メモリ
1400
複数OS管理制御部
第1OS用
CPU群
第2OS用
CPU群
200
136
133
135
134
139
221
211
222
管理端末
端末
端末
230
231
232

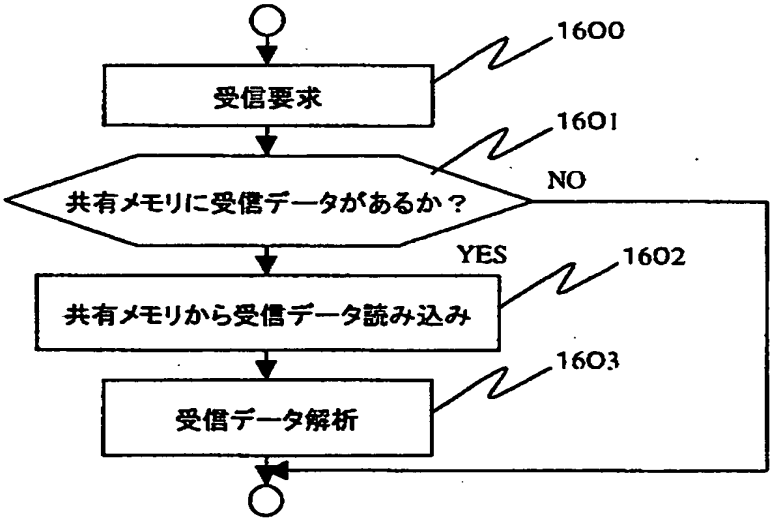
【図１５】

図15



【図１６】

図16



【書類名】　　要約書

【要約】

【課題】　複数のＯＳが用いる計算機資源を、各ＯＳの状況に応じて効率的に割り当て制御する。

【解決手段】　１台の計算機システム内に複数のＯＳを稼動させ、各ＯＳの計算機資源の変更、再配置を行うことが可能にクラスタシステムが構成されている。各ＯＳを運用系、待機系として運用している場合、ＯＳ管理制御部２００は、各ＯＳの状況を監視して、運用系ＯＳの障害時に計算機資源を別の正常なＯＳに多く割り当てて、それを運用系とする。これにより、障害に関わらず、計算機システムの処理能力を変えることなく計算機システムを稼働させることができる。また、ＯＳ管理制御部２００は、各ＯＳの負荷状態を監視して、その負荷に応じて、計算機資源を割り当てることが可能である。これにより、ＯＳ間の処理能力を必要に応じて調整することができる。

【選択図】　　図２

出　願　人　履　歴　情　報

識別番号　　〔０００００５１０８〕

１．変更年月日　　１９９０年　８月３１日
［変更理由］　新規登録
住　所　東京都千代田区神田駿河台４丁目６番地
氏　名　株式会社日立製作所